3-D OBJECT POSITION ESTIMATION AND RECOGNITION BASED ON PARAMETERIZED SURFACES AND MULTIPLE VIEWS

Bruno Cernuschi-Frias,† Peter N. Belhumeur,‡ David B. Cooper,‡

† Facultad de Ingenieria, University of Buenos Aires, Buenos Aires, Argentina

‡ Division of Engineering, Laboratory for Engineering Man/Machine Systems, Brown University, Providence, RI 02912

Abstract

A new approach is introduced to 3-D parameterized object estimation and recognition. Though the theory is applicable for ***any parameterization***, we use a model for which objects are approximated by patches of spheres, cylinders, and planes---primitive objects. These primitive surfaces are special cases of 3-D quadric surfaces. Primitive surface estimation is treated as parameter estimation using data patches in two or more noisy images taken by calibrated cameras in different locations and from different directions. Included is the case of a single moving camera. Though various techniques can be used to implement this nonlinear estimation, we discuss the use of gradient descent. Experiments are run and discussed for the case of a sphere of unknown location. It is shown that the estimation procedure can be viewed geometrically as a cross correlation of **nonlinearly transformed** image patches in two or more images. Approaches to object surface segmentation into primitive object surfaces, and primitive object-type recognition are briefly presented and discussed. The attractiveness of the approach is that maximum likelihood estimation and all the usual tools of statistical signal analysis can be brought to bear, the information extraction appears to be robust and computationally reasonable, the concepts are geometric and simple, and close to optimal accuracy should result.

A. Introduction

Essentially all 3-D object surface estimation from multiple views to date is based on either active stereo using a laser and one or two cameras for triangulation, or on passive stereo involving matching points in two images and using triangulation, or on optical flow. We suggest a new approach in which surfaces of complex objects are locally approximated by 3-D parameterized surfaces, and these parameters are estimated from two or more images taken by calibrated cameras from different locations and directions. Our approach depends on a simple representation for the transformation of one image into another, based on knowledge of camera positions and on the a priori unknown 3-D object parameter values. Estimation accuracy is achieved by processing data in blocks. The actual processing is simple standard statistical signal analysis.

Though this approach, first presented in [1], is completely new as far as we know, it has benefited from previous work, e.g., work by Schalkoff and McVey on tracking [2], and shares some concepts with the recent exciting work of Waxman and Wohn [3].

B. Notation and Description of Camera Motion

Let P be a point in 3-D space and $\mathbf{r}^T = (x\,y\,z)^\dagger$ be its representation in the fixed orthogonal world reference frame. Since we assume that objects do not move, this reference frame is fixed with respect to the objects viewed by the camera, and we will call it the object reference frame (ORF). Let $\mathbf{r}'^T = (x' \;\; y' \;\; z')$ be the representation, of the point P, in the orthogonal reference frame attached to the camera. This reference frame, the camera reference frame (CRF), is such that: (1) The camera optical axis is parallel to the z axis, and it looks at the negative z axis. (2) The x and y axes are parallel to the sides of the image. (3) The origin of the camera reference frame coincides with the center of the image. The image is corrected so that the view is not inverted top to bottom and left to right, i.e., a central projection is used.

Let B denote the 3×3 orthogonal rotation matrix that specifies the three unit coordinate vectors for the CRF in terms of the three unit coordinate vectors for the ORF. Let $\mathbf{r}_c$ specify the origin of the CRF in the ORF. Then

$$\mathbf{r}' = B^T(\mathbf{r} - \mathbf{r}_c), \text{ and } \mathbf{r} = B\mathbf{r}' + \mathbf{r}_c \qquad (1)$$

Note that the rotation matrix B and the translation vector $\mathbf{r}_c$ are assumed to be known.

C. Images of an Object Surface Point in Two Image Frames.

Fig. 1 illustrates the orthographic imaging model used. P denotes a point on a parameterized 3-D surface of interest. This surface is described by a function in the ORF. The function is uniquely determined by specifying the values of a parameter vector $\mathbf{a}$. Point P on the object surface is seen as points having coordinates $\mathbf{s}$ and $\mathbf{u}$ in images 1 and 2, respectively. We assume a ***Lambertian reflectance model***. Then the images of point P at $\mathbf{s}$ and $\mathbf{u}$ will have the same intensity. The techniques proposed will not apply to specular reflectors, without modification, because the location of points on the object surface at which specular reflection occurs depends on the camera location. Since most surfaces of interest are largely Lambertian, the assumption is a useful one. Hence $I_2(\mathbf{u}) = I_1(\mathbf{s})$ where $I_1(\mathbf{s})$ and $I_2(\mathbf{u})$ are the picture functions (image intensity functions) in Frames 1 and 2, respectively.

Though our approach is applicable to any parameterized surface, our interest at present is primarily in planar, cylindrical, and spherical primitive surfaces. The equation of the general quadric surface is (2). The three primitive quadrics of

† A symbol in boldface is a column vector, a superscript capital T attached to a vector denotes vector transpose.

CH2282-2/86/0000/0639$01.00 © 1986 IEEE

interest are obtained by imposing suitable constraints among the coefficients in (2).

$$a_{11}x^2 + 2a_{12}xy + 2a_{13}xz + a_{22}y^2 + 2a_{23}yz + a_{33}z^2 + 2a_{14}x + 2a_{24}y + 2a_{34}z + a_{44} = 0 \qquad (2)$$

Denote $\mathbf{a}^T = (a_{11},\ a_{12},\ldots,\ a_{44})$. In this paper, we take the ORF to be the CRF 1, i.e., the $\mathbf{a}$ vector is that for describing the object surface in the first camera reference frame. Thus, the image at point $\mathbf{s}^T = (x(1),\ y(1))$ in the first image frame is the image of a point having coordinates x(1), y(1), z(1) on the object surface. We see that the picture function at point $\mathbf{s}$ in CRF1 is the picture function at point $\mathbf{u}$ in CRF2, where

$$(\mathbf{u},\ z(2))^T = B^T(\mathbf{r}(1) - \mathbf{r}_c(1)) \qquad (3)$$

Parameters specifying this transformation are three rotation angles specifying the rotation matrix B, and three translation components specifying the location $\mathbf{r}_c(1)$ of the origin of CRF2 in CRF1. Denote this six component parameter vector by $\mathbf{b}$. Denote the functional relationship between $\mathbf{s}$ and $\mathbf{u}$ by (4), with $\mathbf{h}(\mathbf{s},\ \mathbf{b},\ z\,(\mathbf{s},\ \mathbf{a}))$ given by (3).

$$\mathbf{u} = \mathbf{h}(\mathbf{s},\ \mathbf{b},\ z(\mathbf{s},\ \mathbf{a})) \qquad (4)$$

Note $\mathbf{u}$ depends explicitly on $\mathbf{s}$ and z(1), and z(1) is determined by both $\mathbf{s}$ and $\mathbf{a}$ as shown in (2).

D. Estimation of a Using Two Images

If $\mathbf{b}$ is known and $\mathbf{a}$ is $\mathbf{a}_t$, the true $\mathbf{a}$, then (5) holds for each $\mathbf{s}$. Choose a square M×M pixel window in CRF1.

$$I_1(\mathbf{s}) = I_2(\mathbf{h}(\mathbf{s},\ \mathbf{b},\ z(\mathbf{s},\ \mathbf{a}_t))) \qquad (5)$$

Denote this pixel set by D. Consider the error measure (6). Then $e_D(\mathbf{a})$ is a minimum at $\mathbf{a} = \mathbf{a}_t$. Our problem is to estimate $\mathbf{a}_t$ by minimizing (6) with respect to $\mathbf{a}$.

$$e_D(\mathbf{a}) = M^{-2}\sum_{s\epsilon D}[I_1(\mathbf{s}) - I_2(\mathbf{h}(\mathbf{s},\ \mathbf{b},\ z(\mathbf{s},\ \mathbf{a}))]^2 \qquad (6)$$

An interpretation of (5) is that the image $I_2(\mathbf{u})$ can be transformed into the image $I_1(\mathbf{s})$ by a varying scale change that locally consists of a rotation, a nonisotropic stretching, and a translation.

The minimization technique we have used is gradient descent. The gradient of (6) is:

$$\frac{\partial e_D}{\partial \mathbf{a}} = -2M^{-2} \sum_{s\epsilon D}[I_1(\mathbf{s}) - I_2(\mathbf{u})]\frac{\partial I_2(\mathbf{u})}{\partial \mathbf{a}} \qquad (7)$$

with

$$\frac{\partial I_2(\mathbf{u})}{\partial \mathbf{a}} = \left[\frac{\partial I_2(\mathbf{u})}{\partial \mathbf{u}}\right]^T \frac{\partial \mathbf{h}(\mathbf{s},\ \mathbf{b},\ z(\mathbf{s},\ \mathbf{a}))}{\partial z}\frac{\partial z(\mathbf{s},\ \mathbf{a})}{\partial \mathbf{a}} \qquad (8)$$

In general, it is inconvenient to express z as an explicit function of $\mathbf{a}$. Hence, to proceed further denote the left side of (2) by g(x, y, z, $\mathbf{a}$). Then $\frac{\partial z(\mathbf{s},\ \mathbf{a})}{\partial \mathbf{a}} = -\frac{\partial g(x,y,z,\mathbf{a}}{\partial \mathbf{a}} \Big/ \frac{\partial g(x,y,z,\mathbf{a})}{\partial z}$ follows from (2). Putting the preceding together results in (9).

$$\frac{\partial e_D(\mathbf{a})}{\partial \mathbf{a}} = 2M^{-2}\sum_{s\epsilon D}[I_1(\mathbf{s}) - I_2(\mathbf{u})]\left[\frac{\partial I_2(\mathbf{u})}{\partial \mathbf{u}}\right]^T \frac{\partial \mathbf{h}(\mathbf{s},\ \mathbf{b},\ z)}{\partial z} x \left[\frac{\partial g(x,y,z,\mathbf{a})}{\partial \mathbf{a}} \quad \frac{\partial g(x,y,z,\mathbf{a})}{\partial z}\right] \qquad (9)$$

Let B^{13} and B^{23} denote the first two elements of the last column of the 3×3 matrix B^T in (3). Then $\frac{\partial \mathbf{h}(\mathbf{s},\mathbf{b},z)}{\partial z} = (B^{13}\ B^{23})^T$. Hence, computation of (9) involves processing the data to obtain $I_1(\mathbf{s})$, $I_2(\mathbf{u})$, and $\frac{\partial I_2(\mathbf{u})}{\partial \mathbf{u}}$, and computing $(B^{13}\ B^{23})^T$ and the last two partial derivatives from the known camera motion and knowledge of g(x,y,z,$\mathbf{a}$).

A steepest descent algorithm for minimizing (6) is (10).

$$\mathbf{a}_{m+1} = \mathbf{a}_m - \frac{\partial e_D(\mathbf{a}_m)}{\partial \mathbf{a}}\Delta_m \qquad (10)$$

We use a Δ_m that depends on $e_D(\mathbf{a}_m)$ and $\frac{\partial e_D(\mathbf{a}_m)}{\partial \mathbf{a}}$ and has magnitude that goes to 0 as m goes to infinity.

E. An Example: The Sphere

To illustrate the approach, consider a spherical surface described by the equation $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2$, which can be solved explicitly for z, viz, $z = z_0 \pm (R^2 - (x - x_o)^2 - (y - y_o)^2)^{1/2}$. The positive square root is used since the outside surface of the sphere is seen by the camera looking in the negative z direction. Hence, $\left(\frac{\partial z(\mathbf{s},\ \mathbf{a})}{\partial \mathbf{a}}\right)^T = \left(\frac{\partial z}{\partial x_o}\ \frac{\partial z}{\partial y_o}\ \frac{\partial z}{\partial z_o}\ \frac{\partial z}{\partial R}\right)^T =$ $(\ (x - x_o)/(z - z_o),\ (y - y_o)/(z - z_o),\ 1,\ R/(z - z_o)\ )^T$

and $z - z_o = (R^2 - (x - x_o)^2 - (y - y_o)^2\ )^{1/2}$. The vector $\frac{\partial z}{\partial \mathbf{a}}$ can be computed directly from this.

F. Incremental Stereo: Object Estimation from a Sequence of Views

Instead of estimating $\mathbf{a}$ from two frames, assume now that there is a moving camera with known motion and producing a sequence of images $I_1(\cdot)$, $I_2(\cdot)$, ... The camera stops momentarily while an image is taken, though the algorithms can be modified in order that this assumption not be necessary. There are a number of significant advantages in using a sequence of images taken from successive camera locations and directions differing only by small amounts from one to the next. One is that the processing is simpler, since the image data arrays and derived arrays in (12) are all computed at the same $\mathbf{s}$. The price incurred for this is that the effective signal-to-noise ratio for estimating $\mathbf{a}_t$ is then very small when using two images differing little from one another. (The noise present is sensor noise, quantization noise, and other system noise and uncertainty.) However, since many images are available in this incremental stereo approach, a great deal of "dynamic" averaging is possible, and the noise can be effectively combatted.

Let D(n) denote a rectangular MxM pixel data window in the n^th frame. Let $e_{D(n)}(\mathbf{a})$ denote the error $e_{D(n)}(\mathbf{a}) = M^{-2} \sum_{s\epsilon D(n)} [I_n(\mathbf{s}) - I_{n+1}(\mathbf{h}(\mathbf{s},\ \mathbf{c}(n),\ z(\mathbf{s},\ \mathbf{a})))]^2$. We seek the $\mathbf{a}$ that minimizes (11). The parameter vector $\mathbf{c}$(n) specifies the transformation from CRFn to CRF(n+1), which may be time-varying.

$$N^{-1}\sum_{n=1}^{N-1} e_{D(n)}(\mathbf{a}) \qquad (11)$$

Since, $I_n(\cdot)$ and $I_{n+1}(\cdot)$ are assumed not to differ by much, $I_n(\mathbf{s}) = I_{n+1}(\mathbf{u}) \approx I_{n+1}(\mathbf{s}) + \frac{\partial I_{n+1}(\mathbf{s})}{\partial \mathbf{s}}[\mathbf{u} - \mathbf{s}]$. Hence,

$$\frac{\partial e_{D(n)}(\mathbf{a})}{\partial \mathbf{a}} \approx -2M^{-2} \sum_{s \epsilon D(n)} \left\{ I_n(\mathbf{s}) - I_{n+1}(\mathbf{s}) - \left[\frac{\partial I_{n+1}(\mathbf{s})}{\partial \mathbf{s}}\right]^T [\mathbf{h}(\mathbf{s},\ \mathbf{c}(n),\ z(\mathbf{s},\ \mathbf{a})) - \mathbf{s}] \right\} \quad \text{(12)}$$
$$\text{x} \left[\left[\frac{\partial I_{n+1}(\mathbf{s})}{\partial \mathbf{s}}\right]^T \frac{\partial \mathbf{h}(\mathbf{s},\ \mathbf{c}(n),\ z(\mathbf{s},\ \mathbf{a}))}{\partial \mathbf{a}} \right]^T$$

where $\frac{\partial \mathbf{h}(\mathbf{s},\mathbf{c}(n),\ z(\mathbf{s},\mathbf{a}))}{\partial \mathbf{a}}$ is a 2×3 matrix with column j being the partial derivatives of the two components of $\mathbf{h}(\mathbf{s},\mathbf{c}(n),\ z(\mathbf{s},\mathbf{a}))$ with respect to a_j. The beauty of (12) is that it is easier to compute than is (7) because $I_{n+1}(\cdot)$ and $\frac{\partial I_{n+1}(\cdot)}{\partial \mathbf{s}}$ are used at $\mathbf{s}$ rather than at the transformed point $\mathbf{u}$.

Previously, the object surface was defined in CRF1. For this section, the object equation must be transformed to CRFn.

Note that working with (12) requires the storage of 2(N-1) arrays, each M×M, derived from the N images. There are 2 M×M arrays associated with each pair of successive images. This is a great deal of data to store. A compromise is to use $I_n(\cdot)$ to compute $\mathbf{a}_n$ and $\mathbf{a}_{n+1}$ in (10) and to then discard it. Hence, the only data used in computing $\mathbf{a}_{n+1}$ would then be $I_{n+1}(\cdot)$ and $I_n(\cdot)$. If Δ_n goes to 0 at an appropriately slow rate, it is possible to achieve estimation accuracy that is comparable to that achievable by using all of the data---$I_1(\cdot)$,..., $I_{n+1}(\cdot)$--- simultaneously as in (11). This is a "stochastic approximation" type of approach to our parameter estimation problem [6].

G. Algorithm Operation Interpretation

Fig. 2 is useful for illustrating, in two dimensions, the operation of our algorithm for estimating $\mathbf{a}_t$. Spheres in 3-D are shown as circles. Consider the processing of the image patch between points $\mathbf{s}'$ and $\mathbf{s}''$ in Frame 1. This patch is the image of the patch between points $\mathbf{p}'$ and $\mathbf{p}''$ on the true sphere labeled $\mathbf{a}_t$. The same patch on the sphere surface gives rise to the image patch between points $\mathbf{u}'$ and $\mathbf{u}''$ in Frame 2. Now suppose the system's estimation of $\mathbf{a}_t$ is $\mathbf{a}$. The associated sphere is shown. The performance functional for the estimate of $\mathbf{a}$ is given by (6) and is computed as follows. The system thinks that the locations on the sphere surface that give rise to the images at points $\mathbf{s}'$ and $\mathbf{s}''$ in Frame 1 are the intersections of the dashed lines, from $\mathbf{s}'$ and $\mathbf{s}''$, with the sphere labeled $\mathbf{a}$. These sphere surface points would be seen as the images at point $\tilde{\mathbf{u}}'$ and $\tilde{\mathbf{u}}''$ in Frame 2. Hence, the system takes the image patch between points $\tilde{\mathbf{u}}'$ and $\tilde{\mathbf{u}}''$ in Frame 2 and assumes that the image at each point $\mathbf{u}$ in this interval is the same image as the image at a point $\mathbf{s}$ in the interval between $\mathbf{s}'$ and $\mathbf{s}''$ in Frame 1. The points $\mathbf{u}$ and $\mathbf{s}$ are related geometrically as in the figure, or algebraically by (4). Performance functional (6) requires computing this error $I_1(\mathbf{s}) - I_2(h(\mathbf{s},\ \mathbf{b},\ z(\mathbf{s},\ \mathbf{a}))$).

We make the following interesting observations. From the geometry of image formation in Fig. 2, the varying scale change that maps the image patch over interval $[\mathbf{s}'\ ,\ \mathbf{s}'']$ in Frame 1 into the image patch over interval $[\mathbf{u}'\ ,\ \mathbf{u}'']$ is seen. Note that both local scale change and translation are involved in this 2-D illustration.

If the incorrect $\mathbf{a}$ is used in computing the performance functional (6), the patch of image used in Frame 2 is that over the interval $[\tilde{\mathbf{u}}'\ ,\ \tilde{\mathbf{u}}'']$. Note that this interval is both a shift and a varying scaling of the interval $[\mathbf{u}'\ ,\ \mathbf{u}'']$. If instead of a sphere, we were dealing with a planar surface, the scale change would be constant throughout the image.

This interpretation is further illustrated by the following 3-D computer simulations. Figs. 3a,b are two frames illustrating images of the same sphere but taken at different locations and orientations. The angle between the optical axes of the two cameras is 45°. The data was generated by taking a real image with a T.V. camera and projecting the image pattern onto a sphere, from a few different directions. The pattern projected onto the sphere in this way is the pattern that is then viewed by the two cameras to create Frames 1 and 2. Note that all of these projections are done by computer simulation. The image patch used in Frame 1 is the square shown there. Point P is its center point. Points P in Frames 1 and 2 are locations on images of the same point on the 3-D sphere surface. Note how the image in the vicinity of Point P in Frame 2 is a varying scaled transformation of the image in the vicinity of Point P in Frame 1. In Fig. 4, we refer to the six image boxes as 1 thru 6 starting with the upper left and moving left to right and top to bottom. Box 1 is the image shown in the square window in Fig. 3a. The system begins with a guess as to $\mathbf{a}$. This initial guess is shown as that associated with Box 2 in Table 1. Using this $\mathbf{a}$, for each point $\mathbf{s}$ in the window in Fig. 3a the system takes the image at point $\mathbf{h}(\mathbf{s},\ \mathbf{b},\ z(\mathbf{s},\ \mathbf{a}))$ in Fig. 3b and puts this picture function value at location $\mathbf{s}$ in an array. The image so formed is that in Box 2. It is the difference of the picture functions of these two images that enters as $I_1(\mathbf{s}) - I_2(\mathbf{h}(\mathbf{s},\ \mathbf{b},\ z(\mathbf{s},\ \mathbf{a})))$ in (6). Box 3 is the image formed in this way using $\mathbf{a}_1$, the parameter vector following the first iteration of our parameter estimation algorithm. In Boxes 4 and 5 are the transformed images associated with $\mathbf{a}$ values at intermediate stages of the estimation process. Finally, Box 6 is the image associated with the $\mathbf{a}$ value found at the last estimation stage. These $\mathbf{a}$ values for the stages associated with the six boxes are shown in Table 1.

H. Shape of the Error Function

Figures 5 and 6 show the shape of the error function $e_D(\mathbf{a})$ for the experiment described in Sec. G. Fig. 5 shows the graph of the error function produced by holding z_0 and R fixed at the true values and varying x_0 and y_0 over the ranges $-150 < x_0 < 200$ and $-170 < y_0 < 170$, where $\mathbf{a}_t$ is given in table 1. Fig. 6 shows the same error function with x_0 and R held fixed and y_0 and z_0 varied over the ranges $-90 < y_0 < 120$ and $-120 < z_0 < 120$. Note that the error function changes much more rapidly with change in z_0 than with change in x_0 or y_0, where z_0 is the distance from the sphere center to the camera.

I. Maximum Likelihood Estimation of $\mathbf{a}_t$

Maximum likelihood or other standard estimation types can be implemented for $\mathbf{a}_t$. By making and using assumptions concerning the reflected light intensity at the sphere surface, it is possible to get better estimates of $\mathbf{a}_t$. This could be of value when the image data is very noisy. To illustrate these possibilities, we assume the pattern is a fixed but a priori unknown function of $\mathbf{r} = (x,\ y,\ z)^T$ --- points on the sphere surface --- given with respect to CRF1. Let $\mu(\mathbf{r})$ denote this pattern. Then the unknowns to be estimated are $\mathbf{a}_t$ and $\mu(\mathbf{r})$.

$I_n(\mathbf{u})$, the n^th image picture function, is assumed to be a Gaussian r.v. with variance σ^2 and mean $E[I_n(\mathbf{u})] \equiv \mu(\mathbf{s},\ z(\mathbf{s},\ \mathbf{a}_t))$, where $\mathbf{s}$ is such that $\mathbf{u} = \mathbf{h}(\mathbf{s},\mathbf{b}(n),\ z(\mathbf{s},\ \mathbf{a}_t))$. Here, $\mathbf{b}(n)$ specifies the transformation from the ORF to CRFn. Hence, the joint likelihood of $I_n(\mathbf{u})$, $\mathbf{u} \epsilon D(n)$, given $\mathbf{a}$ and $\mu(\mathbf{s},\ z(\mathbf{s},\ \mathbf{a}))$ is

$$(2\pi\sigma^2)^{-M^2}\exp \sum_{\mathbf{u}\epsilon D(n)} \left\{ -(1/2\sigma^2)[I_n(\mathbf{u}) - \mu(\mathbf{s},\ z(\mathbf{s},\ \mathbf{a}))\ \ )]^2 \ \right\} \quad (13)$$

The joint likelihood of $I_n(\mathbf{u})$, $\mathbf{u}\epsilon D(n)$, $n = 1,...,N$, given $\mathbf{a}$ and $\mu(\mathbf{s},z(\mathbf{s},\mathbf{a}))$ is just the product of the expressions (13), and is

$$(2\pi\sigma^2)^{-NM^2}\exp \sum_{n=1}^{N} \sum_{\mathbf{u}\epsilon D(n)} \left\{ -(1/2\sigma^2)[I_n(\mathbf{u}) - \mu(\mathbf{s},\ z(\mathbf{s},\ \mathbf{a})))]^2 \right\} (14)$$

The maximum likelihood estimate of $\mathbf{a}_t$ and $\mu(\mathbf{s},\ z(\mathbf{s},\ \mathbf{a}_t))$ is simply those values for which (14) is a maximum. For $\hat{\mathbf{a}}$, the maximum likelihood estimate of $\mathbf{a}_t$, the maximum likelihood estimate of $\mu(\mathbf{s},\ z(\mathbf{s},\ \mathbf{a}))$ is given by

$$\hat{\mu}(\mathbf{s},\ z(\mathbf{s},\ \hat{\mathbf{a}})) = N^{-1}(\mathbf{s}) \sum_{\substack{n=1 \\ \text{and such that} \\ \mathbf{h}(s,\mathbf{b}(n),z(s,\ \hat{\mathbf{a}}))\epsilon D(n)}}^{N} I_n(\mathbf{h}(\mathbf{s},\ \mathbf{b}(n),\ ,z(\mathbf{s},\ \hat{\mathbf{a}}))\ \ ) \quad \mathbf{(15)}$$

Note that $N^{-1}(s)$ is the number of images in which point $(\mathbf{s},\ z(\mathbf{s},\ \hat{\mathbf{a}}))$ on the sphere surface is seen and is used in the estimation; $\hat{\mu}(\mathbf{s},\ z(\mathbf{s},\ \mathbf{a}))$ is the average of the picture functions at the points assumed to be views of the location $(\mathbf{s},\ z(\mathbf{s},\ \hat{\mathbf{a}}))$ on the sphere surface.

The major difference between estimating $\mathbf{a}_t$ by minimizing (11) and estimating it by maximizing (14) is that in the former case $I_j(\mathbf{h}(\mathbf{s},\ \mathbf{b}(j),\ z(\mathbf{s},\ \hat{\mathbf{a}}))$, $j = n-1,\ n+1$ are each subtracted from $I_n(\mathbf{h}(\mathbf{s},\ \mathbf{b}(n),\ z(\mathbf{s},\ \mathbf{a}))$ and squared in (11), whereas in the latter case the right side of (15) is subtracted from $I_n(\mathbf{h}(\mathbf{s},\ \mathbf{b}(n),\ z(\mathbf{s},\ \hat{\mathbf{a}}))$ and squared in (14).

J. 3-D Surface Type Segmentation, and Primitive Object Type Recognition

Assume the image $I_n(\mathbf{u})$ is a combination of the undistorted image of the 3-D object surface, plus stationary, zero-mean, white Gaussian noise of a priori unknown variance σ^2. Further, assume the noise is independent from one image to the next. The noise model accounts for sensor noise, A/D quantization noise, and other random perturbations or uncertainties in the system. Then $w(\mathbf{s},\ \mathbf{a}_t) \equiv I_1(\mathbf{s}) - I_2(\mathbf{h}(\mathbf{s},\ \mathbf{b},\ z(\mathbf{s},\ \mathbf{a}_t))\ \ )$ is a Gaussian random variable (r.v.) with mean 0 and variance $2\sigma^2$; and the $w(\mathbf{s},\ \mathbf{a}_t)$, $\mathbf{s}\epsilon D$, are i.i.d. r.v.'s, providing

$$\mathbf{h}(\mathbf{s}'\ ,\ \mathbf{b}'\ ,\ z(\mathbf{s}'\ ,\ \mathbf{a}_t)) \neq \mathbf{h}(\mathbf{s}''\ ,\ \mathbf{b},\ z(\mathbf{s}''\ ,\ \mathbf{a}_t)) \qquad (16)$$

for $\mathbf{s}' \neq \mathbf{s}''$ and $\mathbf{s}'\ ,\mathbf{s}'' \epsilon D$. (If $I_2(\mathbf{u})$ is a compression of $I_1(\mathbf{u})$, it is likely that there will be pairs $\mathbf{s}'\ ,\ \mathbf{s}''$ for which (16) is not true. In such a case it is necessary to purge one of these pixels from D for the purpose of the test to be discussed now for testing whether the window is a view of one primitive 3-D object surface or is a view of two or more such surfaces.) When $\mathbf{a} \neq \mathbf{a}_t$, $w(\mathbf{s},\ \mathbf{a})$, $\mathbf{s}\epsilon D$, are not i.i.d. Furthermore, if two or more surfaces are seen within the window in Frame 1, $w(\mathbf{s},\ \mathbf{a})$, $s\epsilon D$, will not be i.i.d.

Let D' denote a subset of D such that all pixels in it satisfy (13), and let L be the number of pixels in D'. Then the $w(\mathbf{s},\ \hat{\mathbf{a}})$ are approximately i.i.d Gaussian r.v.'s where $\hat{\mathbf{a}}$ is the maximum likelihood estimate of $\mathbf{a}_t$. Hence, a test that should be reasonably good for testing the hypothesis that a piece of one primitive object is seen in the image window D' against the alternative that pieces of two or more primitive objects are seen in D' or that $\hat{\mathbf{a}}$ differs considerably from $\mathbf{a}_t$, is a test for the hypothesis that $w(\mathbf{s},\ \mathbf{a})$, $\mathbf{s}\epsilon D'$ are i.i.d. Standard t-tests or F-tests can be used for this purpose. The general quadric surface model can be used and its parameter vector $\mathbf{a}_t$ estimated, or a less careful test can be implemented, once for each of the three models --- sphere, cylinder, plane.

Finally, asymptotically Bayesian recognition is possible by using the type of asymptotic results discussed in [4] and more extensively in [5].

Conclusions

A parametric modeling and statistical estimation approach has been proposed and some simulation shown for estimating 3-D object surfaces from images taken by calibrated cameras in two or more positions. The parameter estimation suggested is gradient descent, though exhaustive search is also possible. Much of the machinery of statistical signal analysis can be brought to bear in our approach. Processing image data in blocks (windows) is central to our approach. An important question of interest is what should these block sizes be in order to achieve acceptable parameter estimation accuracy. The required block size is a function of image noise, 3-D object surface shape, and pattern reflected at the object surface. These and other questions are under study.

Acknowledgement

This work was partially supported by NSF Grant #ESC-81-19676, ARO Grant #DAAG-29-81-K-0167 and by the University of Buenos Aires.

References

[1] B. Cernuschi-Frias, "Orientation and Location Parameter Estimation of Quadric Surfaces in 3-D Space from a Sequence of Images," Ph.D. Thesis, Brown University, Division of Engineering, May 1984. Also Tech. Report #LEMS-5, February, 1984.

[2] R. J. Schalkoff and E. S. McVey, "A Model and Tracking Algorithm for a Class of Video Targets," IEEE Trans. on PAMI, Jan. 1982, pp. 2-10.

[3] A. M. Waxman and K, Wohn, "Contour Evolution, Neighborhood Deformation and Global Image Flow: Planar Surfaces in Motion ," Technical Report, Computer Vision Laboratory, Center for Automation Research, University of Maryland, April 1984.

[4] R. M. Bolle and D. B. Cooper, "Bayesian Recognition of Local 3-D Shape by Approximating Image Intensity Functions with Quadric Polynomials," IEEE Transactions on PAMI, July 1984, pp. 418-429.

[5] R. M. Bolle and D. B. Cooper, "On Optimally Combining Pieces of Information, with Application to Estimating 3-D Complex-Object Position from Range Data," To appear in IEEE Trans. on PAMI; also Technical Report #LEMS-8, Brown University, Division of Engineering, December 1984.

[6] A. E. Albert and L. A. Gardner, **Stochastic Approximation and Nonlinear Regression**, The M.I.T. Press, Cambridge, Mass., 1967.

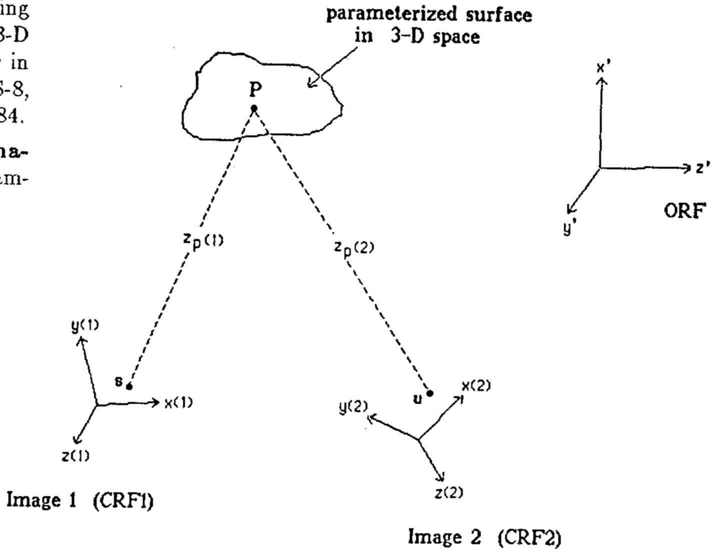

FIGURE 1

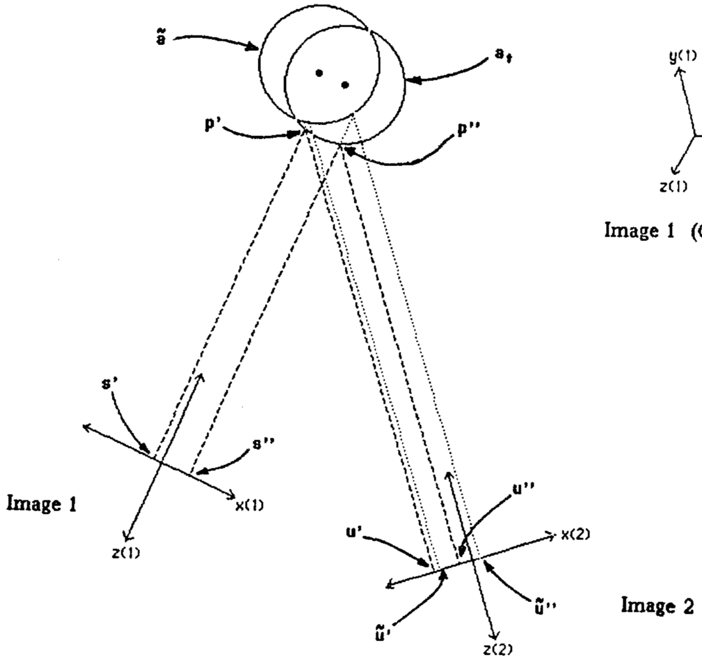

FIGURE 2

Image Box	a			
	x_0	y_0	z_0	R
2	40.00	40.00	-2040.00	128
3	20.83	40.37	-2016.74	128
4	20.79	40.40	-2004.72	128
5	10.52	30.30	-2009.08	128
6	0.41	1.13	-1999.89	128

a_t	0	0	-2000	128
â	0.41	1.13	-1999.89	128

Table 1

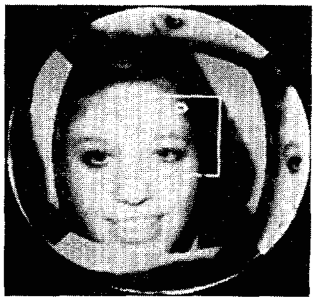

Figure 3a

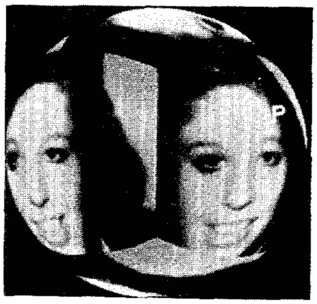

Figure 3b

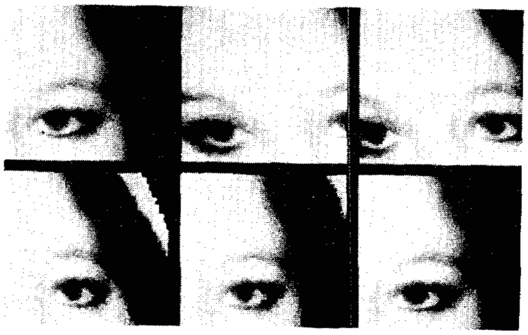

Figure 4

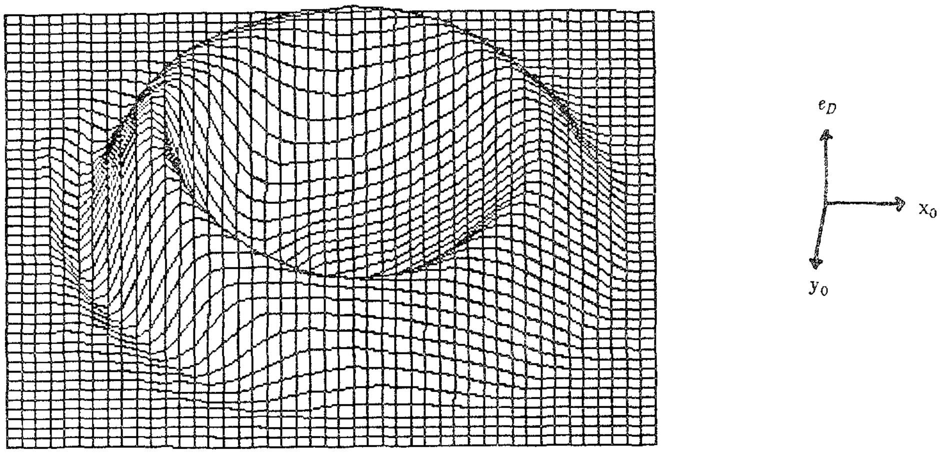

Figure 5

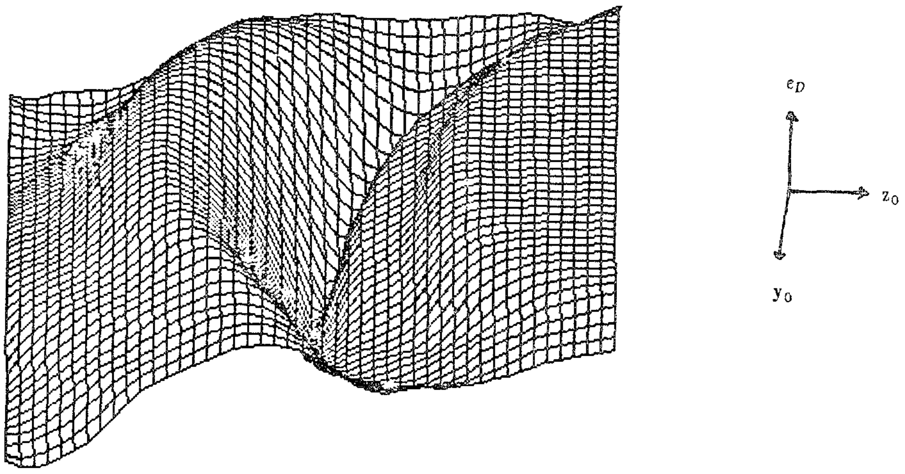

Figure 6